AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer-implemented method for simulating a circuit design for a programmable logic device (PLD), the PLD including two or more types of configurable elements, comprising:

(a) reading a configuration bitstream;

(b) constructing objects in a computer memory, each object corresponding to a configurable element of the PLD as configured in the configuration bitstream, and each object having associated therewith an output signal state and one or more input signal states;

(c) generating events in response to signal values in the configuration bitstream, each event including an object identifier, an input signal identifier, and an input signal state;

(d) for each event, updating the output signal state and an input signal state of a corresponding object in response to the input signal state of the event and type of configurable element;

(e) performing steps (f) - (g) [[(h)]] if processing an event changes the output signal state of an object;

(f) finding the configurable elements that are connected to the output signal; and

(g) generating events for the objects corresponding to the configurable elements from step (f).

2. (Currently Amended) The method of claim 1, wherein the PLD is a field programmable gate array (FPGA), and further comprising:

constructing lookup table objects corresponding to FPGA lookup tables, wherein each lookup table object includes a plurality of input signal attributes and an ordered set of bit values, each bit value addressable by values of the input signal attributes.

3. (Original) The method of claim 1, wherein each event further includes a routing delay value.

4. (Original) The method of claim 1, further comprising:

adding each event to a last-in-first-out (LIFO) queue; and

getting each event from the LIFO queue prior to processing the event.

5. (Original) The method of claim 4, further comprising maintaining a synchronous event LIFO queue for synchronous events and an asynchronous event LIFO queue for asynchronous events.

6. (Original) The method of claim 5, further comprising:

maintaining respective synchronous event queues for different clock periods;

maintaining respective asynchronous event queues for different clock periods; and

for new events generated in processing a current event, selecting synchronous and asynchronous event queues for the new events as a function of accumulated signal delay values.

7. (Original) The method of claim 6, wherein the PLD is a field programmable gate array (FPGA), and further comprising:

constructing lookup table objects corresponding to FPGA lookup tables, wherein each lookup table object includes a plurality of input signal attributes and an ordered set of bit values, each bit value addressable by values of the input signal attributes.

8. (Original) The method of claim 6, wherein each event further includes a routing delay value.

9. (Original) The method of claim 4, further comprising selectively discarding events.

10. (Original) The method of claim 9, wherein each object further includes one or more timestamps associated with the one or more input signal states, and each event includes a timestamp, the method further comprising:

storing time values as timestamps for input pins of objects, each time value being a function of a time at which an event was processed for an object;

storing time values as timestamps for events when events are generated, each time value for an event being a function of a time at which an event was generated;

discarding an event if the timestamp of the event is less than the timestamp of the input pin of the object.

11. (Original) The method of claim 10, wherein the PLD is a field programmable gate array (FPGA), and further comprising:

constructing lookup table objects corresponding to FPGA lookup tables, wherein each lookup table object includes a plurality of input signal attributes and an ordered set of bit values, each bit value addressable by values of the input signal attributes.

12. (Original) The method of claim 10, wherein each event further includes a routing delay value.

13. (Original) The method of claim 1, further comprising, in finding the configurable elements that are connected to the output signal as an event is processed, generating objects corresponding to the configurable elements.

14. (Original) The method of claim 1, further comprising simulating the circuit design in response to hardware interface program calls that are selectable for interfacing with a physical device or with a simulation process.

15. (Original) The method of claim 14, wherein the PLD is a field programmable gate array (FPGA), and further comprising:

constructing lookup table objects corresponding to FPGA lookup tables, wherein each lookup table object includes a plurality of input signal attributes and an ordered set of bit values, each bit value addressable by values of the input signal attributes.

16. (Original) The method of claim 14, wherein each event further includes a routing delay value.

17. (Original) The method of claim 14, further comprising:

adding each event to a last-in-first-out (LIFO) queue; and

getting each event from the LIFO queue prior to processing the event.

18. (Original) The method of claim 17, further comprising maintaining a synchronous event LIFO queue for synchronous events and an asynchronous event LIFO queue for asynchronous events.

19. (Original) The method of claim 18, further comprising:

maintaining respective synchronous event queues for different clock periods;

maintaining respective asynchronous event queues for different clock periods; and

for new events generated in processing a current event, selecting synchronous and asynchronous event queues for the new events as a function of accumulated signal delay values.

20. (Original) An apparatus for simulating a circuit design for a programmable logic device (PLD), the PLD including two or more types of configurable elements, comprising:

means for reading a configuration bitstream;

means for constructing objects in a computer memory, each object corresponding to a configurable element of the PLD as configured in the configuration bitstream, and each object having associated therewith an output signal state and one or more input signal states;

means for generating events in response to signal values in the configuration bitstream, each event including an object identifier, an input signal identifier, and an input signal state;

means for updating the output signal state and an input signal state of a corresponding object, for each event, in response to the input signal state of the event and type of configurable element;

means for finding the configurable elements that are connected to the output signal if processing an event changes the output signal state of an object;

means for generating events for the objects corresponding to the configurable elements connected to the output signal if processing an event changes the output signal state of an object.